



Московский Государственный Университет имени М.В. Ломоносова

Факультет вычислительной математики и кибернетики

Отчет по заданию №2 в рамках курса
«Суперкомпьютерное моделирование и технологии»
Численное интегрирование многомерных функций
методом Монте-Карло

Выполнил:

студент группы 608

Канзепаров Денис Ринатович

Вариант № 9

Москва, 2022

Содержание

1. Математическая постановка задачи	2
2. Численный метод решения	2
3. Нахождение точного значения интеграла аналитически	3
4. Краткое описание прогораммной реализации	3
5. Исследование мастшабируемости программы на системе Polus.	4

1. Математическая постановка задачи

Функция $f(x, y, z)$ – непрерывна в ограниченной замкнутой области $G \subset \mathbb{R}^3$. Требуется вычислить определенный интеграл:

$$I = \int \int \int_G f(x, y, z) dx dy dz.$$

Интегрируемая функция:

$$f(x, y, z) = xy^2z^3,$$

область G ограничена поверхностями $z = xy$, $y = x$, $x = 1$, $z = 0$.

2. Численный метод решения

Пусть область G ограничена параллелепипедом Π :

$$\begin{cases} a_1 \leq x \leq b_1, \\ a_2 \leq x \leq b_2, \\ a_3 \leq x \leq b_3. \end{cases}$$

Рассмотрим функцию $F(x, y, z)$:

$$F(x, y, z) = \begin{cases} f(x, y, z), & (x, y, z) \in G \\ 0, & (x, y, z) \notin G. \end{cases}$$

Преобразуем искомый интеграл:

$$I = \int \int \int_G f(x, y, z) dx dy dz = \int \int \int_{\Pi} F(x, y, z) dx dy dz.$$

Пусть $p_1(x_1, y_1, z_1), p_2(x_2, y_2, z_2), \dots$ – случайные точки, равномерно распределенные в Π . Возьмем n таких случайных точек. В качестве приближенного значения интеграла предлагается использовать выражение:

$$I \approx |\Pi| \cdot \frac{1}{n} \sum_{i=1}^n F(p_i), \quad (1)$$

где $|\Pi|$ – объем параллелепипеда Π . $|\Pi| = (b_1 - a_1)(b_2 - a_2)(b_3 - a_3)$.

3. Нахождение точного значения интеграла аналитически

Вычислим интеграл аналитически:

$$\begin{aligned} I &= \int_0^1 \int_0^x \int_0^{xy} xy^2 z^3 dx dy dz = \int_0^1 \int_0^x xy^2 \frac{(xy)^4}{4} dx dy = \frac{1}{4} \int_0^1 \int_0^x x^5 y^6 dx dy = \frac{1}{4} \int_0^1 x^5 \frac{x^7}{7} dx = \\ &= \frac{1}{28} \int_0^1 x^{12} dx = \frac{1}{28} \frac{1}{13} = 0.00(274725) \approx 0.002747252747 \end{aligned}$$

Далее в качестве точного решения данного интеграла будем использовать значение 0.002747252747.

4. Краткое описание прогормамной реализации

Реализуем парадигму «мастер–рабочие»: один из процессов («мастер») генерирует случайные точки и передаёт каждому из остальных процессов («рабочих») отдельный, предназначенный для него, набор сгенерированных случайных точек. Все процессы–рабочие вычисляют свою часть суммы по формуле (1). Затем вычисляется общая сумма с помощью операции редукции.

«Мастер»: проверяет условие сходимости, если оно не выполнено, то продолжает работу, а именно, увеличивает число создаваемых точек в 2 раза, определяет новое зерно для вычисления случайных координат точек, создает массив с координатами точек. Далее начинает рассылать данный массив по «рабочим».

«Рабочий»: проверяет условие сходимости, если оно не выполнено, то продолжает работу, а именно, принимает массив случайных точек, суммирует значения функции в точках, которые лежат в области G , вычисляет значение интеграла по формуле (1). Далее высылает полученный результат на «мастер».

«Мастер»: складывает полученные с «рабочих» результаты, проверяет верность условия сходимости и рассылает по «рабочим» значение переменной, отвечающей за прекращение вычислений, если достигнута точность. Если точность достигнута, то выводится результат работы программы. Иначе делает еще одну итерацию.

Таблица 1. Таблица с результатами расчётов для системы Polus

Точность ε	Число MPI-процессов	Время работы программы (с)	Ускорение	Ошибка
$3.0 \cdot 10^{-5}$	2	0.0114360789	1	0.0000063418
	4	0.0059280460	1.9291481375	0.0000254376
	16	0.0050992691	2.2426898199	0.0000094070
	32	0.0100861250	1.1338426700	0.0000180734
$5.0 \cdot 10^{-6}$	2	0.7285899210	1	0.0000014995
	4	0.6601575660	1.1036606388	0.0000043206
	16	0.6359944200	1.1455916877	0.0000032540
	32	0.7432084291	0.9803305405	0.0000031747
$1.5 \cdot 10^{-6}$	2	0.6942877981	1	0.0000009065
	4	0.3303793331	2.1014867715	0.0000005525
	16	0.3610963470	1.9227217441	0.0000012337
	32	0.7447988479	0.9321816220	0.0000011781

5. Исследование масштабируемости программы на системе Polus.

Мы видим, что реализация парадигмы «мастер–рабочие» для данной задачи не дает значительного ускорения, а в некоторых случаях, даже ухудшает производительность. Это объясняется тем, что время затраченное на пересылки увеличивается с увеличением количества MPI-процессов. А вычислений становится меньше на каждом из MPI-процессов.

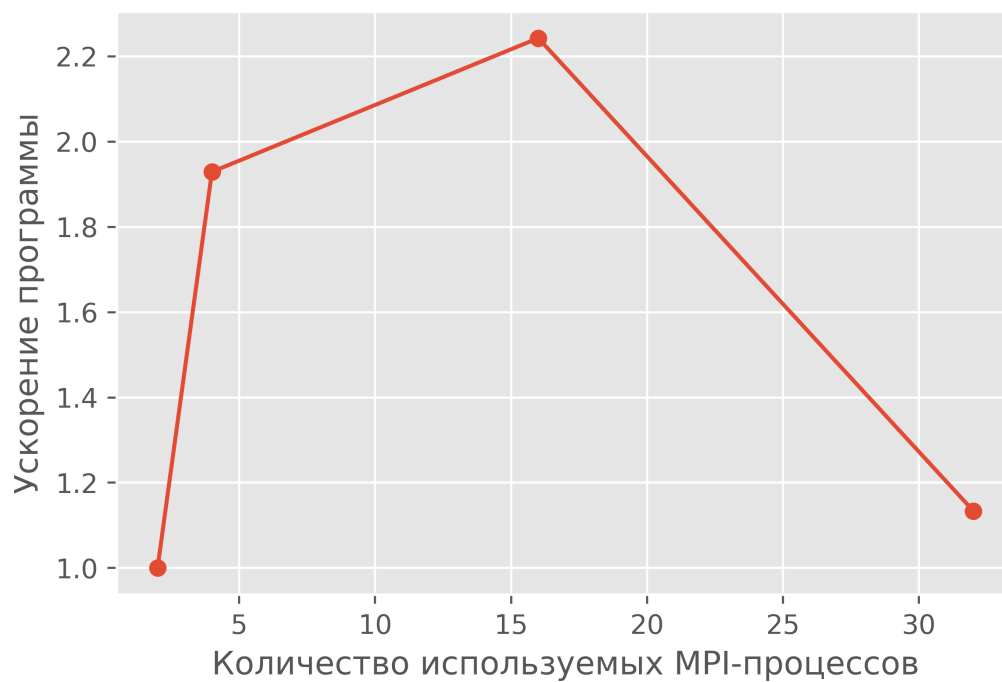


Рис. 5.1. График зависимости ускорения программы от числа используемых MPI-процессов для значения $\varepsilon = 0.00003$

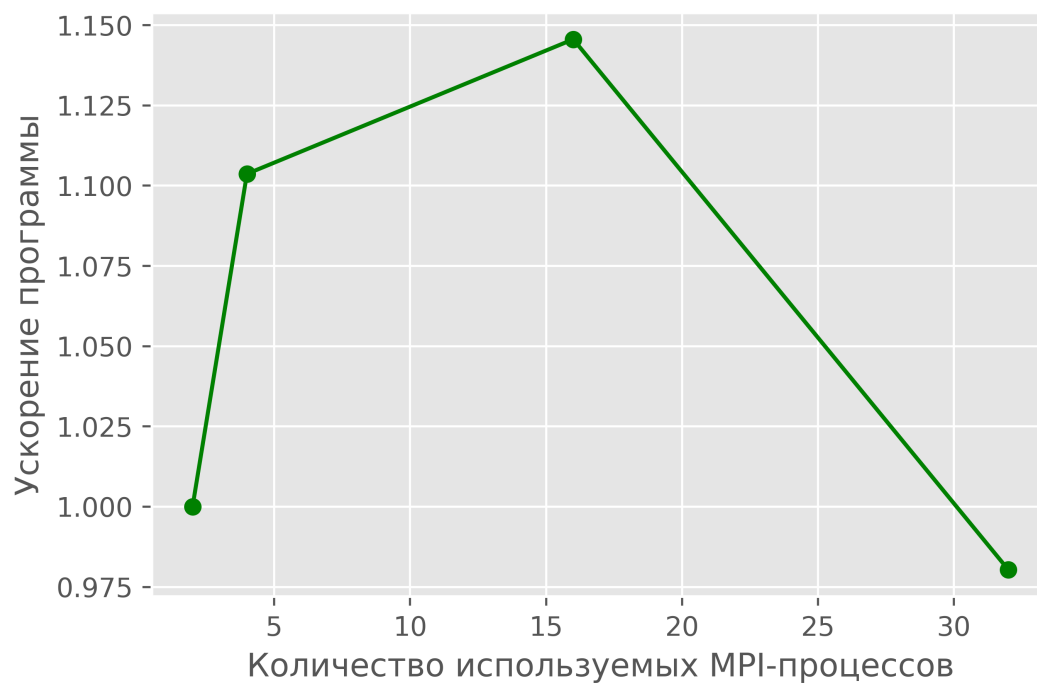


Рис. 5.2. График зависимости ускорения программы от числа используемых MPI-процессов для значения $\varepsilon = 0.000005$

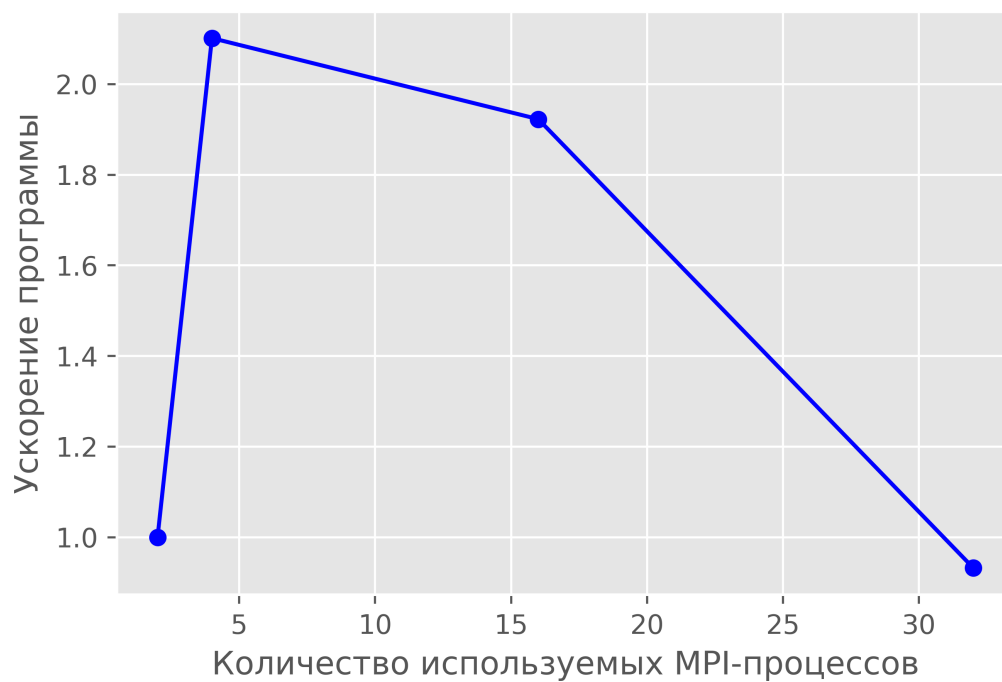


Рис. 5.3. График зависимости ускорения программы от числа используемых MPI-процессов для значения $\varepsilon = 0.0000015$

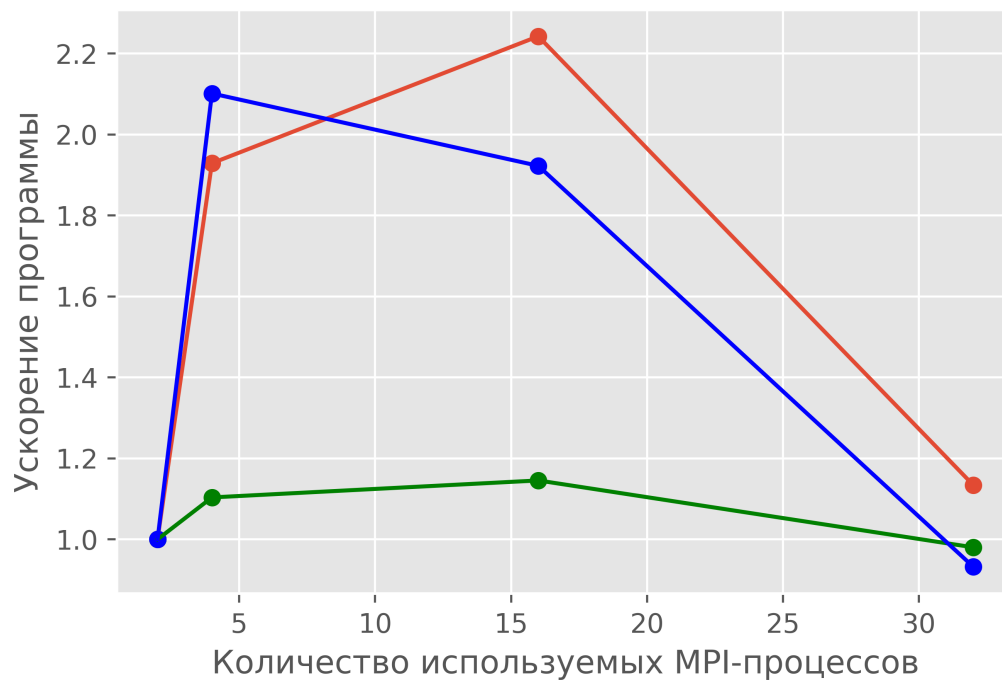


Рис. 5.4. Графики зависимости ускорения программы от числа используемых MPI-процессов

Таблица 2. Таблица с результатами расчётов для системы Polus для фиксированного зерна

Точность ε	Число MPI-процессов	Время работы программы (с)	Ускорение	Ошибка
$3.0 \cdot 10^{-5}$	2	0.0239540620	1	0.0000158589
	4	0.0136748330	1.7516895453	0.0000289734
	16	0.0135930070	1.7622342135	0.0000188537
	32	0.0193841921	1.2357524046	0.0000296541
$5.0 \cdot 10^{-6}$	2	0.3476793999	1	0.0000025991
	4	0.1678119920	2.0718388224	0.0000027915
	16	0.1237788299	2.8088761234	0.0000029932
	32	0.4208524390	0.8261313650	0.0000035649
$1.5 \cdot 10^{-6}$	2	5.6767938490	1	0.0000011178
	4	2.0902041229	2.1014867715	0.0000012182
	16	2.8145470200	2.0169475971	0.0000005053
	32	5.3075983659	1.0695598004	0.0000008992

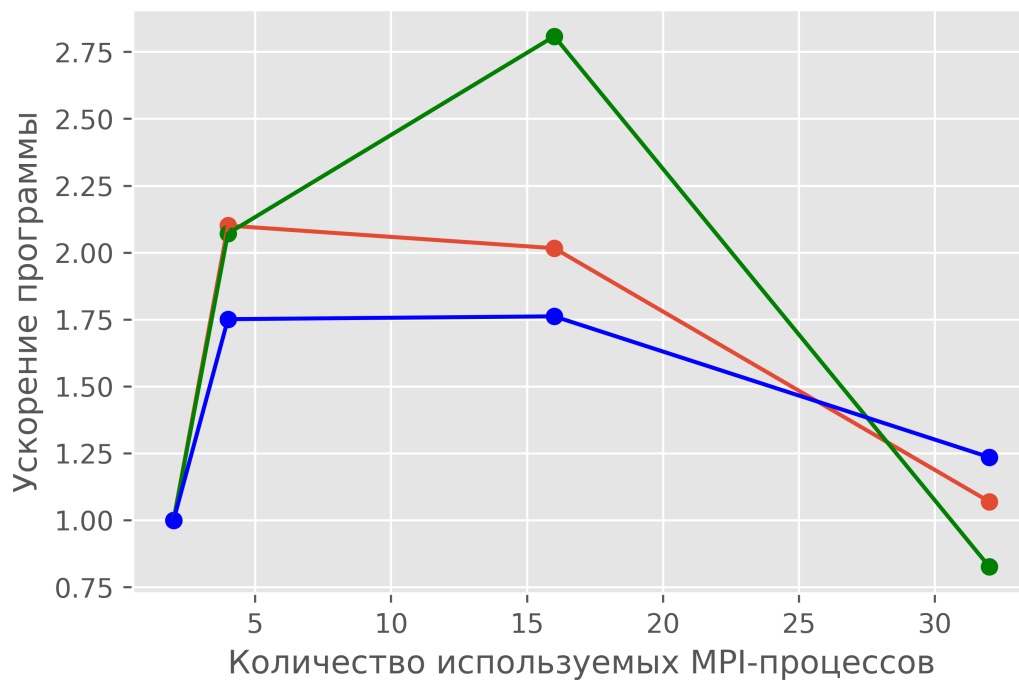


Рис. 5.5. Графики зависимости ускорения программы от числа используемых MPI-процессов